Unsupervised Learning: Letting Data Speak for Itself

Ahmed BADI
ahmedbadi905@gmail.com
linkedin.com/in/badi-ahmed

December 17, 2025

Abstract

Unsupervised learning is the part of machine learning that works without labels. Instead of predicting a known output, the goal is to discover structure hidden inside the data itself: natural groups of points, lower-dimensional manifolds, or recurring patterns and co-occurrences. This article provides a clear and intuitive tour of unsupervised learning, focusing on three core families of methods: clustering, dimensionality reduction, and association rule learning. We start from everyday analogies, then move gradually to the mathematical formulations behind popular algorithms such as k-means, hierarchical clustering, Principal Component Analysis (PCA), t-SNE, and Apriori [1], [2], [3], [4], [5]. Along the way, we discuss anomaly detection and modern representation learning as important extensions of the unsupervised mindset [6], [7]. With equations, figures, and practical examples, this guide aims to demystify unsupervised learning while keeping the style simple and human-friendly.

Keywords: Unsupervised Learning, Clustering, Dimensionality Reduction, PCA, t-SNE, Association Rules, Apriori, Anomaly Detection.

1 Introduction

Imagine entering a party where you know nobody. There are no name tags, no labels saying "Data Scientist", "Engineer" or "Artist". Yet, after a few minutes of observation, you start to notice patterns: a group in the kitchen talking about football, a cluster near the window discussing cameras, and a couple of people alone, checking their phones.

Without any explicit labels, your brain is already doing **unsupervised learning**: grouping similar people, identifying outliers, and compressing the scene into a mental summary.

In machine learning, unsupervised methods operate in the same way. We are given a dataset $X = \{x_1, \ldots, x_n\}$, where each $x_i \in \mathbb{R}^d$ is a feature vector, but there is no target label y_i attached. The goal is not to predict, but to **discover**:

- Groups or **clusters** of similar points.
- Lower-dimensional structure that explains most of the variability.
- Frequent co-occurrences and patterns such as "people who buy X also buy Y".
- Unusual points that do not fit any discovered pattern (anomalies).

Classic textbooks such as Hastie et al. and Bishop provide a rigorous overview of these tasks and algorithms [1], [2].

In this article, we will focus on three main pillars of unsupervised learning:

1. **Clustering**: grouping similar data points.

2. **Dimensionality Reduction**: compressing high-dimensional data into fewer meaningful dimensions.

3. **Association Rules**: discovering "if–then" relationships in large transactional datasets.

We will also briefly touch on anomaly detection and modern representation learning (autoencoders, self-supervised learning) as natural extensions.

2 Clustering: Discovering Groups

Clustering tries to partition data into K groups such that points in the same group are more similar to each other than to points in other groups [1].

2.1 k-means Clustering

k-means is one of the simplest and most widely used clustering algorithms [3]. Given a desired number of clusters K, it tries to find K centroids $\{\mu_1, \ldots, \mu_K\}$ that minimize the within-cluster sum of squares:

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \, \|x_i - \mu_k\|^2, \tag{1}$$

where $r_{ik} \in \{0, 1\}$ indicates whether point x_i belongs to cluster k.

The algorithm alternates between two simple steps:

1. **Assignment step**: assign each point to the nearest centroid:

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_j \|x_i - \mu_j\|^2 \\ 0 & \text{otherwise} \end{cases}$$

2. **Update step**: recompute each centroid as the mean of its assigned points:

$$\mu_k = \frac{\sum_{i=1}^{n} r_{ik} x_i}{\sum_{i=1}^{n} r_{ik}}.$$

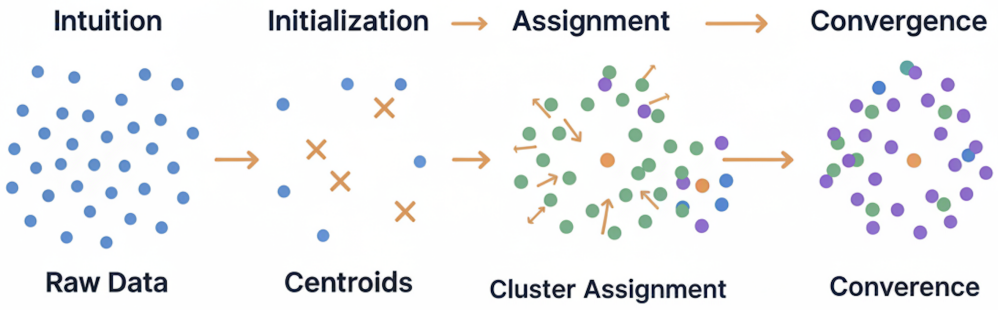

Figure 1: k-means clustering: initialization, assignment, and centroid update over iterations.

Despite its simplicity, k-means works surprisingly well when clusters are roughly spherical and of similar size. It struggles, however, with non-convex clusters or clusters of very different densities [1].

2.2 Hierarchical Clustering

Hierarchical clustering builds a tree of clusters, called a **dendrogram** [2]. In the *agglomerative* (bottom-up) version:

- Start with each point as its own cluster.
- Iteratively merge the two closest clusters according to a linkage criterion (single, complete, average).
- Continue until only one cluster remains.

Cutting the dendrogram at different heights gives different numbers of clusters, which is useful for exploratory data analysis.

2.3 Density-Based Clustering (DBSCAN)

k-means assumes clusters are compact and roughly spherical. **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) takes a different view: a cluster is a region of high point density, and points in low-density regions are labeled as noise or outliers [8]. Two key parameters are:

- ε: neighborhood radius.
- `minPts`: minimum number of points required to form a dense region.

DBSCAN can find arbitrarily shaped clusters and is robust to noise, but requires tuning these parameters.

3 Dimensionality Reduction: Compressing Data

Modern datasets often have hundreds or thousands of features: pixels in images, words in documents, sensors in IoT. High dimensionality makes visualization, computation, and learning harder (the **curse of dimensionality**) [1].

Dimensionality reduction aims to transform $x \in \mathbb{R}^d$ into $z \in \mathbb{R}^q$ with $q \ll d$ while preserving as much information as possible [2].

3.1 Principal Component Analysis (PCA)

PCA is the workhorse of linear dimensionality reduction [9]. It finds directions (principal components) that capture the maximum variance in the data.

Given centered data matrix $X \in \mathbb{R}^{n \times d}$ (each column has mean 0), the sample covariance matrix is

$$\Sigma = \frac{1}{n} X^\top X. \tag{2}$$

PCA computes eigenvalues and eigenvectors of Σ:

$$\Sigma v_j = \lambda_j v_j, \tag{3}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$.

The first q eigenvectors $\{v_1, \ldots, v_q\}$ define a q-dimensional subspace that captures the largest possible variance. The projection is

$$z_i = V_q^\top x_i, \tag{4}$$

where $V_q = [v_1, \ldots, v_q]$.

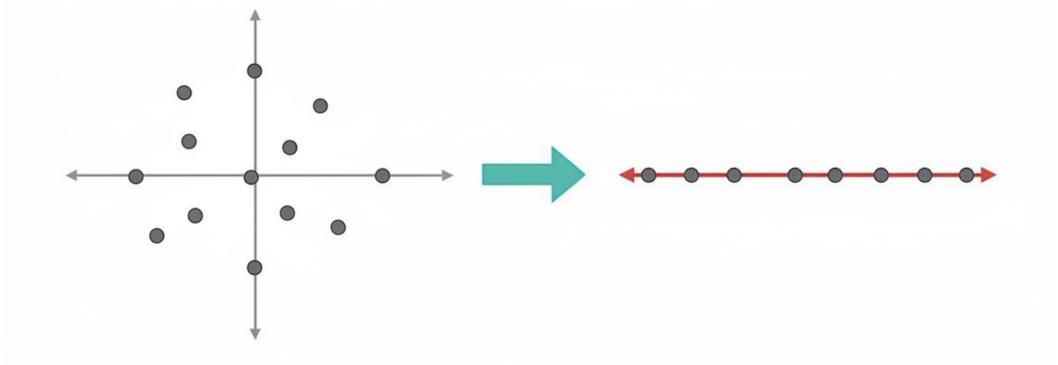

Figure 2: PCA: data in 2D projected onto the first principal component (maximum variance direction) [9].

The proportion of variance explained by the first q components is

$$\text{Explained Variance Ratio} = \frac{\sum_{j=1}^{q} \lambda_j}{\sum_{j=1}^{d} \lambda_j}. \tag{5}$$

3.2 Nonlinear Manifold Learning: t-SNE and UMAP

PCA is linear. For complex data (images, text embeddings) the true structure may lie on a curved low-dimensional manifold. **t-SNE** (t-distributed Stochastic Neighbor Embedding) builds a low-dimensional map that preserves local neighborhoods [4]:

- In high dimension, define pairwise similarities p_{ij} using a Gaussian kernel.
- In low dimension, define similarities q_{ij} using a Student-t distribution.
- Optimize positions $\{z_i\}$ to minimize the Kullback–Leibler divergence:

$$\mathrm{KL}(P \parallel Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

t-SNE is excellent for visualization (2D plots of clusters), but not ideal as a generic feature extractor because distances are not preserved globally.

UMAP (Uniform Manifold Approximation and Projection) is a more recent technique that builds a graph in high dimension and optimizes a similar graph in low dimension, typically preserving more global structure while remaining fast and scalable [10].

4 Association Rules: "If–Then" Patterns

Association rule learning discovers relationships between items in large transactional datasets. The classic example is **market basket analysis** in retail [5].

Each transaction T is a set of items (e.g., {Bread, Butter, Milk}). An association rule has the form

$$X \Rightarrow Y,$$

where X and Y are itemsets and $X \cap Y = \emptyset$. For example:

$$\{\text{Diapers}\} \Rightarrow \{\text{Beer}\}$$

can be read as "customers who buy diapers also tend to buy beer".

Three key measures quantify the strength of a rule:

- **Support**: how often the full itemset appears:

$$\text{supp}(X \cup Y) = \frac{\#\text{transactions containing } X \cup Y}{\#\text{all transactions}}.$$

- **Confidence**: how often the rule is true when the left side occurs:

$$\text{conf}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}.$$

- **Lift**: how much more often X and Y occur together compared to independence:

$$\text{lift}(X \Rightarrow Y) = \frac{\text{conf}(X \Rightarrow Y)}{\text{supp}(Y)}.$$

4.1 Apriori Algorithm

The Apriori algorithm systematically searches for frequent itemsets using the **downward closure** property: if an itemset is frequent, all its subsets are also frequent [5].

High-level steps:

1. Find all frequent single items (support above a threshold).
2. Use them to generate candidate itemsets of size 2, 3, and so on.
3. Prune candidates whose subsets are not frequent.
4. From frequent itemsets, generate rules with high confidence and lift.

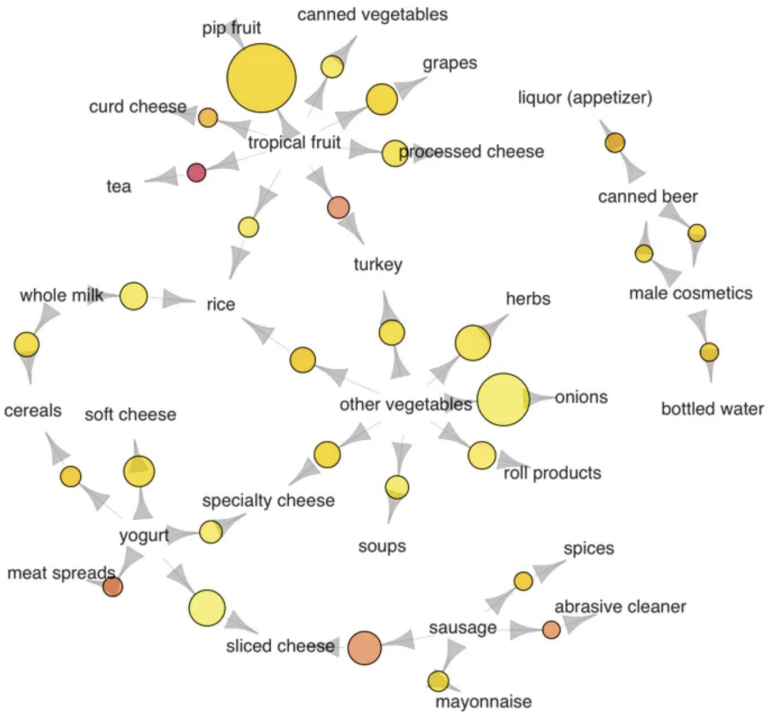

Figure 3: Association rule mining: from transactions to frequent itemsets and association rules [5].

Image source:
`https://miro.medium.com/v2/resize:fit:1400/format:webp/1*BEFF9cZF92Xn9Frds3XD4g.png`

Other algorithms such as FP-growth and Eclat improve efficiency by avoiding explicit candidate generation [1].

5 Beyond the Basics: Anomalies and Representations

5.1 Anomaly Detection

An anomaly (or outlier) is a data point that does not conform to the expected pattern. In many applications, anomalies are the most interesting points: fraudulent credit card transactions, intrusions in network traffic, or defective parts in manufacturing.

Unsupervised anomaly detection assumes that normal data is frequent and lies in dense regions, while anomalies are rare and lie in sparse regions [1]. Popular methods include:

- Distance-based (points far from neighbors are anomalies).
- Density-based such as LOF (Local Outlier Factor) [6].
- One-Class SVM and Isolation Forest.

5.2 Representation Learning and Autoencoders

Modern unsupervised learning often focuses on learning good **representations** of data. Autoencoders are neural networks trained to reconstruct their input:

$$x \xrightarrow{\text{Encoder}} z \xrightarrow{\text{Decoder}} \hat{x}. \tag{6}$$

By constraining z to be low-dimensional or sparse, the autoencoder learns a compressed representation similar in spirit to nonlinear PCA [2], [7].

These latent vectors z can then be used for visualization, initialization for supervised tasks, or clustering in representation space.

6 Practical Considerations

6.1 Preprocessing and Scaling

Many unsupervised methods rely on distances (k-means, DBSCAN, PCA), so feature scaling is crucial. Common approaches:

- Standardization: $x' = (x - \mu)/\sigma$.
- Min–max scaling to $[0, 1]$.

6.2 How Many Clusters? How Many Dimensions?

For clustering, there is no ground truth K. Heuristics include the elbow method and the silhouette score, both of which are widely discussed in the literature [1]. For PCA, choose q such that the explained variance ratio exceeds a threshold (e.g., 90%) [9].

6.3 When to Use What?

- Use **k-means** for simple, spherical clusters and speed.
- Use **DBSCAN** when you expect arbitrary shapes and noise.
- Use **PCA** for linear compression and as a preprocessing step.
- Use **t-SNE/UMAP** mainly for 2D/3D visualization.
- Use **association rules** for transaction data and recommendation.

7 Advantages and Limitations

7.1 Advantages

1. **No Labels Needed**: Can extract value from raw, unlabeled data, which is far more common than labeled data [1], [2].
2. **Exploratory Power**: Reveals hidden structures, segments, and relationships that were not known in advance.
3. **Preprocessing Tool**: Dimensionality reduction and clustering can improve downstream supervised models.

7.2 Limitations

1. **No Ground Truth**: It is often hard to objectively evaluate results without labels.
2. **Parameter Sensitivity**: Many algorithms require hyperparameters (K, ε, perplexity) that significantly affect results.
3. **Interpretability**: Low-dimensional embeddings and clusters may be hard to interpret without domain knowledge [2].

8 Conclusion

Unsupervised learning is about curiosity: asking the data what stories it wants to tell, instead of forcing it into predefined labels. Through clustering, we discover natural groupings; through dimensionality reduction, we compress complexity into simpler views; through association rules, we uncover patterns of co-occurrence that drive recommendation engines and business insights [1], [2].

In practice, unsupervised learning rarely acts alone. It is often the first step in a larger pipeline: exploring new datasets, engineering features, cleaning anomalies, and preparing inputs for supervised models. Understanding these techniques not only makes us better machine learning practitioners, but also better data explorers.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, 1967.

[4] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[5] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 487–499, 1994.

[6] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 93–104, 2000.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*, vol. 1, pp. 318–362, 1986.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Proceedings of KDD*, pp. 226–231, 1996.

[9] I. Jolliffe, "Principal component analysis," *Springer Series in Statistics*, 2002.

[10] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," in *Proceedings of the 2018 Workshop on Statistical Techniques in Pattern Recognition*, 2018.